

Internet Gopher

The Bridge to the Web

Bob Alberti

Robert Alberti

June 1, 2011

Internet Gopher:
The Bridge to the Web
Bob Alberti

Introduction

In April of 1990 I took a job as a programmer/help desk operator with the University of Minnesota's Microcomputer Workstation and Network Center, then part of the Academic Computing Systems and Services department (which has since gone through half a dozen name changes to become the Office of Information Technology.) I heard from an acquaintance that this was quiet, interesting work, and the Regent's Scholarship employee benefit program offered free classes that might allow me to complete my undergraduate degree (yes, the same undergraduate degree for which this paper is written, twenty years later!) After only a few months of the low-stress, low-demand job I had been led to expect by my acquaintance I was placed on a project to write something called "Internet Gopher" (Gopher).

Rapid Growth

The Internet had by 1991 reached a point of information supersaturation: there were approximately 375,000 Internet hosts in the world, up from 80,000 only two years prior, with the number of hosts was increasing logarithmically (Lottor, 1992).

Already rapid, commercial involvement in the Internet would further increase its growth. However, the Internet was still at this time running primarily across a backbone funded by the National Science Foundation (the NSFnet backbone), and commercial exploitation of this public infrastructure was prohibited in most cases (NSFnet, 1992). The quarter million hosts with domain names ending in ".edu" (indicating colleges, universities, and other educational institutions) still outnumbered all others including commercial hosts ending in ".com."

Some method was needed to index and organize the data on these systems, both on the macro scale of the Internet and on the scale of individual institutions. The University of Minnesota was among many exploring the usefulness of computers as means to convey information to the campus – the campus-wide information system, or CWIS.

Enter the Gopher

The project to which I had been assigned employed new “client-server” software architecture, and I was responsible for a lot of C-language UNIX coding, including the Unix Gopher client. Working with client-server architecture was fascinating, although its overall impact was by no means immediately clear to me. We were writing Gopher in response to the University of Minnesota’s Campus-Wide Information Services (CWIS) committee, about which my boss Mark McCahill had very little good to say. Apparently this committee had been meeting for several years, and while it had assembled a phonebook-thick collection of functional requirements specifying to a very fine degree of detail what the application should do, no actual code had yet been written.

McCahill’s plan, as explained after I attended my first confusing meeting of the CWIS committee in November of 1990, was to create a working prototype CWIS application in the month before the next scheduled meeting, and to debut it there. The committee that had not produced a line of code in years would be treated to a complete client-server application created in-between meetings. I was naïve enough to think such an accomplishment would be lauded by the committee.

Development was intense, including 36-hour programming sessions (Romenesko, 1996).

[McCahill] with assistance from the newly formed Gopher Team, fine-tuned the prototype as music from Nirvana and Mudhoney blared in the background.

"It was a fun time," recalls team member Torrey. "It was a lot of late nights and weekends and a lot of beer, pizza and speed metal."

A month later we had working examples of a Gopher server, and Mac and UNIX clients. I attended my second and last CWIS committee meeting, where we demonstrated the capabilities of the newly-created “Gopher” program.

While I naïvely expected that our hard work would be well-regarded, the reception was instead what might be expected when a number of civil service bureaucrats are presented with the surprise of an accomplishment that bypasses all of their efforts, and threatens to render all their work moot. Gopher team member Farhad Anklesaria called the meeting a “total disaster” (Frana, 2004). I remember one committee member literally jumping up and down in fury. At the end of the meeting, the CWIS committee prohibited the Gopher programming team from any further development on the application.

Since we were prohibited from working on Gopher, we put the source code up on the FTP server on boombox.micro.umn.edu (Boombox) and informed colleagues at other institutions about its availability. Remember in those days the only way to retrieve something from the Internet was to know its address in advance, so the only way for information about the availability of Gopher's source code to spread via Usenet conferencing (Anklesaria F. , 2011), e-mail discussion lists or verbally.

Outsourced

Despite the challenge of getting word out, the prototype Gopher client and server proved compelling enough to attract attention. In part this was due to the fact that one needed almost nothing to run a Gopher client. By providing a public login over Telnet (Figure 1) individuals who were curious about Gopher could immediately try out the UNIX client, and this demonstration was often sufficient to persuade them to download and install the Gopher software via Xmodem or FTP transfer.

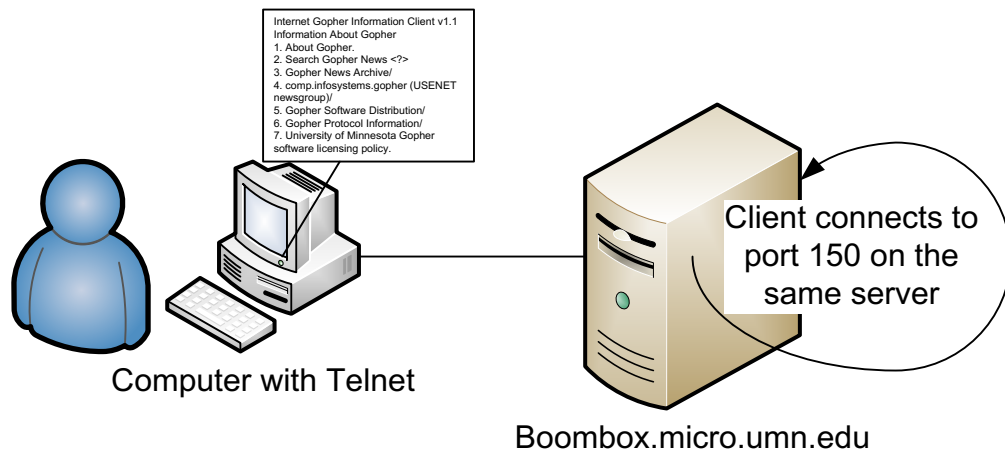


Figure 1 Telnet to Gopher Client on Boombox

Distribution was quite rapid once people were aware Gopher existed. By facilitating its own distribution, Gopher may have been the Internet's first "viral application."

The Gopher team members were not shy about explaining to those who wrote with suggestions for improvements that we were prohibited from working on Gopher by the campus CWIS group, and we happily provided the names of University administrators to whom comments might be addressed. Senior University personnel began receiving calls and e-mails from other institutions complimenting them on Gopher and asking about the Gopher development prohibition. After a couple of months, Gopher's popularity and undeniable utility became an embarrassment to the CWIS committee, and we were able to formally resume the work we had continued doing on our own time.

History

When I entered the University of Minnesota in 1980 computers were already employed for registration, although the method was crude by modern standards. Each seat in a course was represented by a computer punch-card encoded with the appropriate registration information. To register one proceeded through the established paper-based registration processes making numerous trips on foot between the financial offices, the university registration offices, and the individual college and program offices to gain the appropriate approvals. Once approved the student was handed the computer card that represented their seat in the course, and made a final trip over to the university registration office, where the card was handed to a clerk to be added to the day's registration run.

When the stack of the day's cards was batch-read into the mainframe registration computer, one's registration was finally complete.

This mainframe legacy was part of the cultural divide that made my visits to the CWIS committee so very exciting. The committee was primarily composed of those whose whole experience of computers was limited to mainframes, and who were accustomed to the slow pace of technologic change in mainframe technology.

McCahill's team was younger: programmer Paul Lindner had only just graduated from college, I was 29, and McCahill was 34. All of us had experienced the rapid development of personal computers (PCs) during the

1970's and 1980's. My first computer had

been a PDP 8/L (Figure 2) that booted from toggle switches in 1977. My first laptop was a NEC-PC8201a (Figure 3) which I carried to class in 1983 to looks of sheer astonishment at its 8-line, 40-character display. In six years I'd gone from wire-wrapped core memory to something not unlike a Neo Dana (Figure 4) available today.



Figure 3 NEC PC8201a



Figure 2 PDP 8L loading 0201 with 6622 (Skip)

McCahill, Anklesaria and Lindner were particularly interested in client-server architecture, which distributed the work of a given computational tasks between two computers: the server responds tersely to many requests for data; the client issues requests, receives responses, and handles the computationally-large work of displaying the results to the user. Nowadays client-server architecture is ubiquitous, but in 1991 the growth of the



Figure 4 Neo Dana

Internet (e.g. servers) and the increase in power of the personal computer (clients) had developed to the point where client-server architecture was increasingly feasible. And as PCs became more and more graphic-dependent, there was no option from the point of view of computing power but to offload management of those graphics to the client systems.

Unfortunately the University of Minnesota campus CWIS committee was unfamiliar with and skeptical of client-server architecture. From their point of view the mainframe was an established means of computing, and they had little reason to believe personal computers would be useful as anything other than toys. The Gopher team, on the other hand, had developed a feel for Moore's Law (Moore, 1965), and understood (from such experiences as mine with my laptop) that personal computers would soon outstrip the power of established mainframe systems. Client server architecture leveraged this power to distribute the computing load across servers and clients, allowing for a much richer user experience than could possibly be managed by a mainframe somehow running all displays.

While it is easy to criticize the campus CWIS committee for its failure of vision, it's important to note that if they had not restricted the development of Gopher by its U of MN authors, Gopher may not have been so rapidly distributed and popularized across the Internet. Interestingly, this paralleled my experience a few years prior with my company GamBit MultiSystems (Alberti, 2010): if a former employee had not cut us off from our original market (p. 37), we would not have responded by franchising to thirteen cities in the U.S. and Canada. Being "forced from the nest" under unfortunate circumstances led to greater eventual success in both cases.

Setting

In order to understand Gopher's significance and its impact on contemporary computing in 1991, it is important to understand the environment from which it emerged. In 1991, computers were the realm of academics and hobbyists, and the landscape of services and connections was much more fractured and difficult to navigate than it is today. Connectivity was primarily provided by modems with speeds ranging from 300 to 2400 baud (Daxial Communications, 2003). E-mail was granular within institutions to the level of individual departmental mail server – you couldn't write to "name@institution.edu," and there were no on-line directories. Most interpersonal contact was by

reading Usenet, which was an increasingly unwieldy¹ database of interest-based forums distributed via NNTP protocol (Kantor & Lapsley, 1986) to a growing number of servers around the Internet.

The primary means of moving files was over File Transfer Protocol (FTP) (Postel, 1980). FTP's stateful architecture and unusual two-port communications protocol is an artifact of its antiquity. FTP was developed back when there were no personal computers, only mainframe computers and dumb terminals, or maxi-Hosts and TIPs respectively in the original ARPANET design (Edmondson-Yurkanan, 2002). The Dumb Terminal was used to tell a Source Mainframe to exchange files with Target Mainframe (Figure 5). The Source Mainframe had to talk to the Dumb Terminal and the Target Mainframe on separate communications lines, since it had to maintain connection state with both of them.

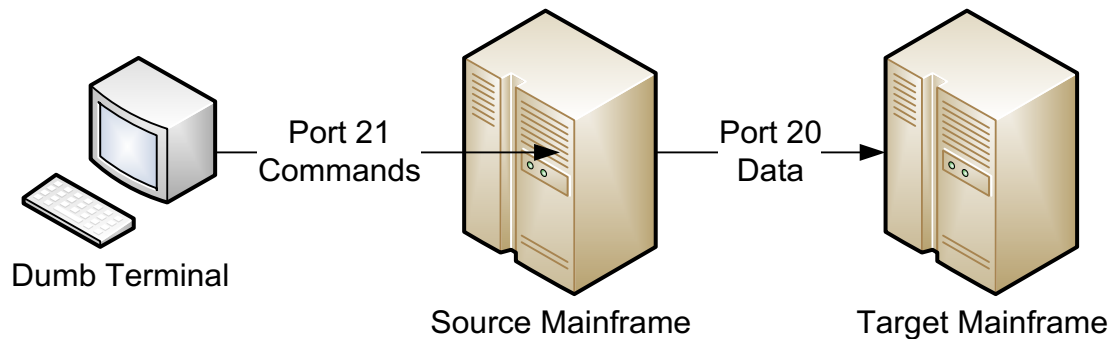


Figure 5 Original FTP architecture

When PC's emerged, FTP had to be adapted accommodate the new paradigm. The "Target Mainframe" was now actually the hard drive on the PC (Figure 6), resulting in the FTP architecture most commonplace today.

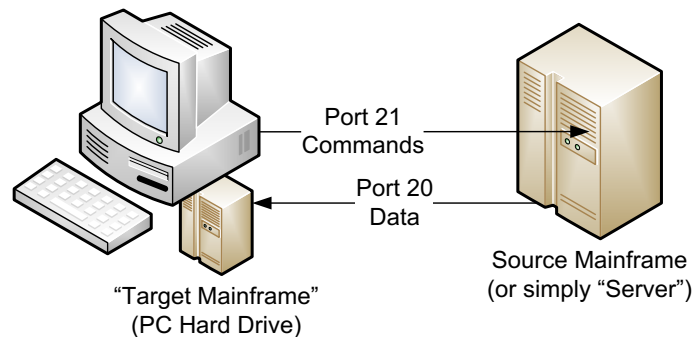


Figure 6 FTP on PC architecture

¹ I was a Usenet administrator from 1994-1996 and the network load imposed by this service was ridiculous. Since there was no subscription mechanism only a tiny fraction of Usenet data was ever read from a given server.

As the Internet grew, firewalls were used to protect private networks, rendering the FTP architecture even more problematic. The Firewall, a device designed to *exclude* arbitrary inbound connections had to be taught how to *permit* Port 20 FTP data inbound connections, and how to route those to the various client PCs which had requested the connection (Figure 7). So the Firewall must note outbound FTP requests, and maintain a table of internal client PC's to which inbound connection attempts might be expected at some future time. Such “stateful inspection” firewalls had not been invented when Gopher was being written (Higgins, 2008).

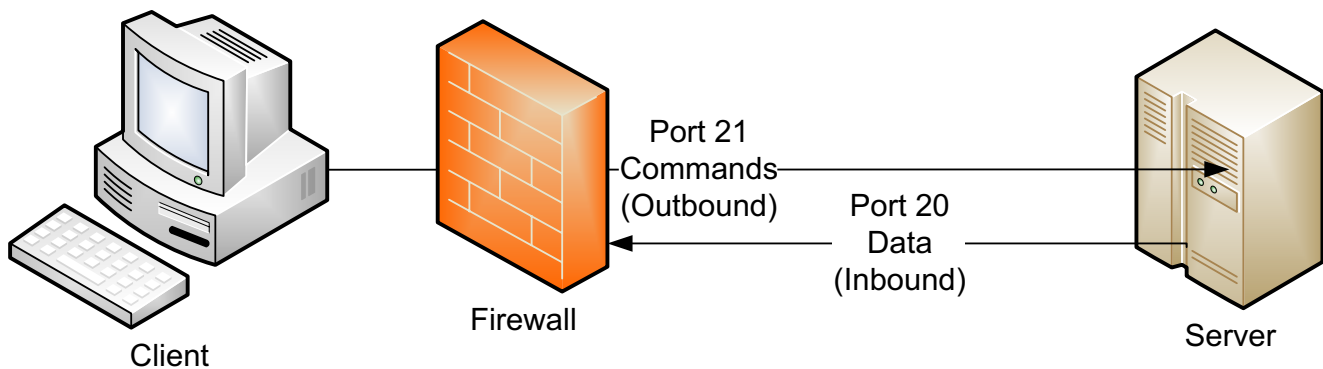


Figure 7 FTP through Firewall

To be sure, this is a simplification that illustrates FTP in “Active” mode. There is also a “Passive” mode which allows limited bidirectional communications on Port 21. There are also issues regarding the transmission of binary versus ASCII data. Active and Passive, Binary and ASCII, all of these were additional complications that made FTP a challenge to many early Internet adopters. In order to download a file from a server through a firewall, you had to put your computer in Binary mode, then Passive mode, then initiate the transfer all while maintaining a connected session state. Maintaining the connected session state could be a problem when downloading large files, as some FTP servers would time out if no new commands were received in a certain period. Depending on the transmission speed (remember, there were still 300 baud modems in use), the time required to transfer a file could exceed the maximum connection time!

Modern FTP software has addressed these challenges by writing smarter, more powerful client software that would have been impossible back when we were developing Gopher. All of these settings had to be carefully and deliberately configured, and lists of useful FTP sites and their particular settings had to be maintained by hand.

Finally, FTP service required a login. By 1991, an informal but well-understood convention had been adopted of anonymous FTP login using one of the accounts “ftp,” “guest,” or “anonymous,” with the individual’s e-mail address as an untested password (the authenticity of the e-mail address was unverified). A limit on the number of simultaneous anonymous logins meant that popular FTP sites were frequently full and not immediately accessible.

Structure of the Gopher protocol

Notable about Gopher is its extremely terse communication protocol, written to support the “lowest common denominator” protocols, which in 1991 included 300-baud modems. Gopher provides acceptable performance over slow connections with the protocol described in Figure 8.

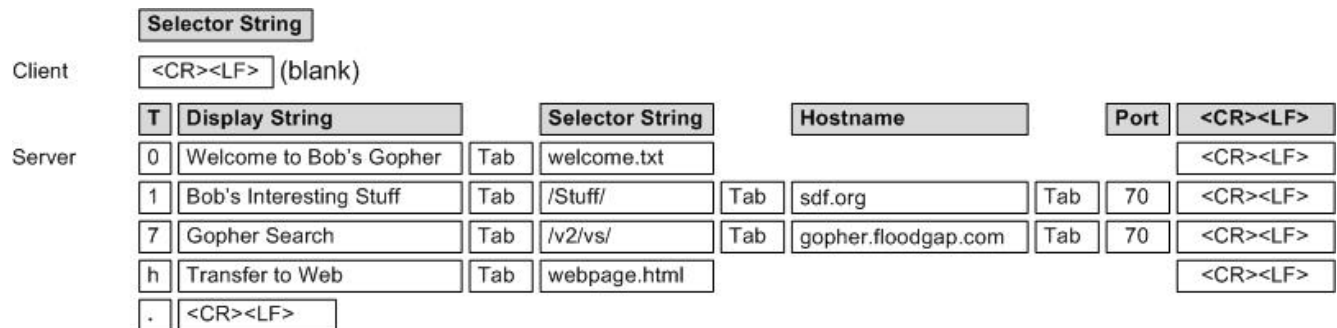


Figure 8 Basic Gopher Protocol

The protocol is described in the gray boxes in Figure 8 (which are not part of the transmission):

Type Display String (tab) Selector String (tab) hostname (tab) port (end of line)

Unusually for a protocol, Gopher mixes column-delimited (the single initial type character) and tab-delimited fields (all the others).

Developed before the invention of the Universal Resource Locator (URL) which has since become a mainstay of Web communications, the Gopher protocol can be seen to contain the elements that became part of the URL specification in 1994. In fact both Gopher’s McCahill and the Web’s Berners-Lee are co-authors of the URL definition RFC 1738 (Berners-Lee, Masinter, & McCahill, 1994). It is now possible to use a single-line URL such as <Gopher://sdf.org:70/users/alberti/welcome.txt> to reach the same document.

The initial TYPE character notifies the client as to the type of data available at the location specified by the other fields.

The Display String is the user-friendly label behind which the technical information leading to the item will be hidden. This innovation was an important contribution towards Gopher's initial popularity, as it facilitated the use of the Internet by "normal" users.

The Selector String is the system-specific path to the individual item. This is the element most responsible for making the Gopher link transparent to the end user. By properly structuring the Selector String, resources could be obtained from a variety of different kinds of servers without the user needing to know anything about those servers.

The Hostname and Port specify the TCP/IP addressable location of the Internet service that can respond to the request specified in the Selector String. Some of the characteristics that emerge from this protocol are explored below.

Stateless

The Gopher protocol was originally designed to be extremely spare, due in large part to the restrictions on modem speeds that existed when it was written. Most protocols at the time were *stateful*, meaning that one initiated a transaction session, conducted a number of transactions, and disconnected when finished. For example, the FTP protocol, which at the time was the nearest analog of Gopher, is begun with a connection to the FTP server and continues in connected state until a QUIT command is issued or the connection is broken, at which point the state returns to disconnected.

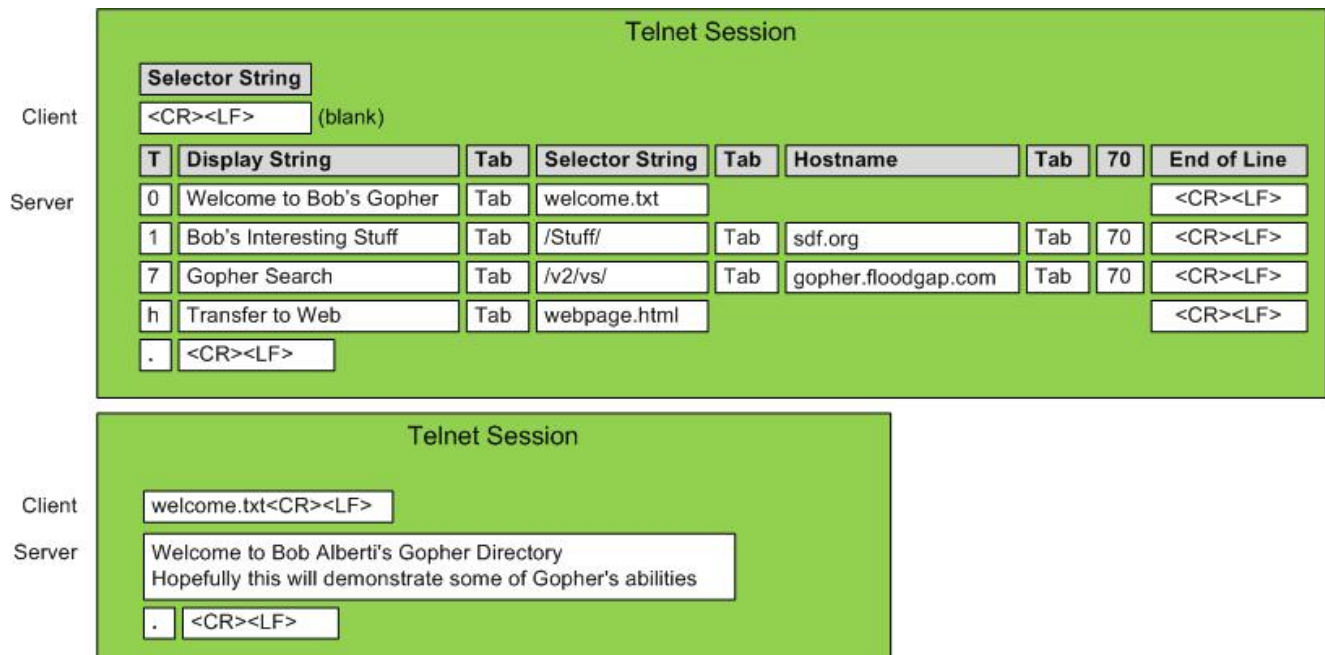


Figure 9 Gopher's Stateless protocol

Novel for its time, the Gopher protocol is *stateless*, that is, each communication from the client to the server is considered atomic, and no state is retained between requests. In Figure 9, each large box is a separate Telnet session: since the Gopher protocol is stateless, sessions can occur in any order.

Links

Additionally, FTP had no means to refer users to other FTP locations, and this was a critical difference between Gopher and FTP. To search for items of various sorts, each FTP site had to be individually visited – login, set modes, manually traverse directory trees, etc. Gopher allowed resources in multiple locations and accessed via multiple protocols – including FTP – to be dynamically assembled into a list of items that could be examined with a single click apiece.

Modes

As previously described, as well as States, FTP had many modes: Active and Passive, Binary and ASCII. This was because FTP could not interpret meta-information about a document. For example, if a filename ends in TXT, people and contemporary FTP software clients know to place the FTP session in ASCII mode: if the filename ends in EXE, binary mode. But the FTP protocol itself cannot make this determination.

Gopher links begin with a single character that describes the type of file which is linked, allowing the protocol itself to inform the client what kind of file is being transferred – what in FTP would be ‘setting the mode.’ This is equivalent to the MIME type (Freed & Borenstein, 1996).

Anonymous

Internet Gopher protocol was anonymous, requiring no login and further simplifying the user experience. Anonymous Telnet access to Gopher clients was a function of the server, not the protocol. The Gopher+ protocol (about which more later) forms allowed for authentication, but this was not a requirement of the protocol, rather a function of the application to which the protocol was applied.

Abstraction

By including meta-information such as file type and location behind a user-friendly Display String Gopher *abstracts* the data away from its underlying technology, allowing users to access data on the Internet without knowing anything about the data source. This is the important and most critical distinction between Gopher and its predecessors, and a major reason, along with performance, for its

initial popularity. Abstraction allowed for a point-and-click interface by placing the work of determining file type, transmission mode, server type, location, etc. into the client software with via the elements we have already examined.

The Heyday of Gopher

The overall impact of the Gopher architecture cannot be overstated – abstracted data access and fast performance made Gopher significantly more user-friendly than anything that had yet been seen. And its deliberately lean client-server design allowed for an acceptable user experience even on computers employing connections as slow as 300 baud.

Crystallizing the Solution

Gopher dropped like a seed crystal into the supersaturated information solution of the Internet, and over the next two years gained broad and enthusiastic acceptance, particularly among computer experts as well as information scientists (colloquially, “librarians”) who sought to ensure that Gopher facilitated formal information sciences methods and notations (Dalton, 1991). By 1993 Internet Gopher escaped the communities of computer mavens and librarians and emerged into popular culture, aided by books such as “The Whole Internet User’s Guide and Catalog,” (Krol, 1992) which helped introduce regular users to the tools necessary to explore the Internet.

For the Gopher programming team, the excitement of working on such an important project never wore off, but the reality of supporting an application under the scrutiny of the entire world was increasingly daunting. It was quickly apparent that the University of Minnesota, while it enjoyed the notoriety brought to it by the protocol named for its mascot, had no interest in actually supporting the project financially. Arguably, a contemporary institution finding itself in possession of something as notable as Gopher might allocate resources to support the innovation, but the University of Minnesota was in the midst of crisis involving its computing resources.

Challenges

In 1991, then Vice-President for Academic Affairs Ettore Infante attempted to privatize all computer services at the University of Minnesota (Dawson, 1991), meaning that during its first six months the development of Internet Gopher was carried out under the pall of imminent layoffs. The University was not going to increase the size of the Gopher team even as it was planning to terminate all computer science employees.

The worry about these plans reached such elevated levels that one IT staff member actually bugged the University of Minnesota Regents conference room in order to learn what was being planned. A computer used for displaying reports had been added to the Regent's conference room, and one IT staffer (who shall remain anonymous even 20 years later) wrote a very simple program to digitize the input from the computer's audio port, and route it to his desktop computer. He then left the computer running in the conference room, with its screen deactivated. Several of us would gather around his desktop to listen to everything that was said during these meetings.²

The Gopher programming team members were never relieved of any of their original duties (Riddle, 1993).

"Paul Lindner spoke about the trials and tribulations of getting a CWIS off the ground at UMinn. Although he and the rest of the Gopher development team are celebrities in Gopherspace, he's "just another joe" at his home institution."

Like everyone else on the Gopher team, I wrote code unrelated to Gopher as well, including SLIPDIAL (Figure 10), a utility for establishing TCP/IP connections over a dialup modem. And I conducted numerous consulting assignments to various University departments, including moving the School of Journalism from Token Ring to Ethernet cabling. All of us worked about 12 hours a week on the walk-in and telephone computer consulting that was the basic function of the department. And during the summer sessions I was part of a team that ran a computer camp for high school students.

```

/* LOGIN.C      Robert M. Alberti, Jr.      February, 1991
/* SLIPDIAL.C   Robert M. Alberti, Jr.      July, 1992
/*
/* X.00 Serial driver package written by Raymond L. Gwinn
/*
/* X.00 driver package distributed with permission of the author.
*/

#include <stdio.h>           // INCLUDES
#include <stdlib.h>
#include "slipdial.h"
#include "x00.h"

FILE *fgLog;                // GLOBAL VARIABLES
FILE *fgProc;               /* Log file */
FILE *fgConfig;             /* Modem command file */
/* CONFIG.TEL */

int iGBaud, iGParity, iGPort; /* Linkage of DTR with SetBaud command */

```

Figure 10 SLIPDIAL.C

On top of our basic work, plus our work supporting Gopher, and our concerns for our jobs, I was also under a unique degree of stress in 1991 because we were expecting our first children, twins. Due in October, my wife was placed on hospitalized bed rest for premature labor in June, three weeks after we moved into our new house. She was hospitalized until the twins were delivered in August, two

² Honestly it didn't amount to much. I learned that major decisions are merely presented for votes that were settled before the meeting. However this helped pique my interest in computer security, resulting in a career change five years later.

months premature. Twenty years later they are fine and healthy, but my stress levels at the time were truly incredible.

I describe all of this to indicate that the circumstances under which Gopher was developed and maintained were far from ideal. Without the voluntary support efforts of contributors from around the Internet, there is no way that Gopher, and later Gopher+, could have succeeded. And it was this lack of support from the University of Minnesota, and our over-reliance upon voluntary community development support, that helped contribute to Gopher's eventual decline.

Publicity

Balancing all of these challenges were the rewards of working on a cutting-edge project. In 1992 the Gopher team hosted the first of several invitation-only GopherCons, attended by fifty people that first year. Gopher broke through to the popular consciousness following a write-up in the London Guardian in August of 1993 (Flowers, 1993). A LexisNexis search for “Internet Gopher” turns up over a dozen articles in 1993 and 1994 published in such diverse periodicals as the Washington Post (Williams, 1995), The Age (Melbourne) (Watson & Barry, 1995), the Business Times of Singapore (Leong, 1994), and Newsweek (Watson & Barry, 1995).



Figure 11 MTV VJ Adam Curry in Gopher T-shirt

By January of 1994, MTV VJ (video jockey, as opposed to “disc jockey”) Adam Curry agreed to wear an Internet Gopher T-shirt on the air (Figure 11) in exchange for MTV receiving a commercial Gopher server license.

Gopher+

Gopher+ introduced a number of additions at the request of the Gopher user community, and in response to features that had been recently added to HTML. These extra features were enabled by tacking a plus (“+”) onto the end of the Gopher protocol (Figure 12). Earlier implementations of Gopher would supposedly ignore extra data beyond the end of the prior protocol (in practice, of course, some clients and servers would crash and had to be rewritten).

Depending on the values after the “+”, the Gopher+ client and server could engage in different negotiations about subsequent communications. While too detailed to fully explore in this paper, there are two basic themes: 1) Attributes, used to collect more information about the item, possibly to request a customized version of the item (such as documents presented in a user’s preferred language); 2) Interactive Query Items to request the client send information to the server in response to a form.

Attributes

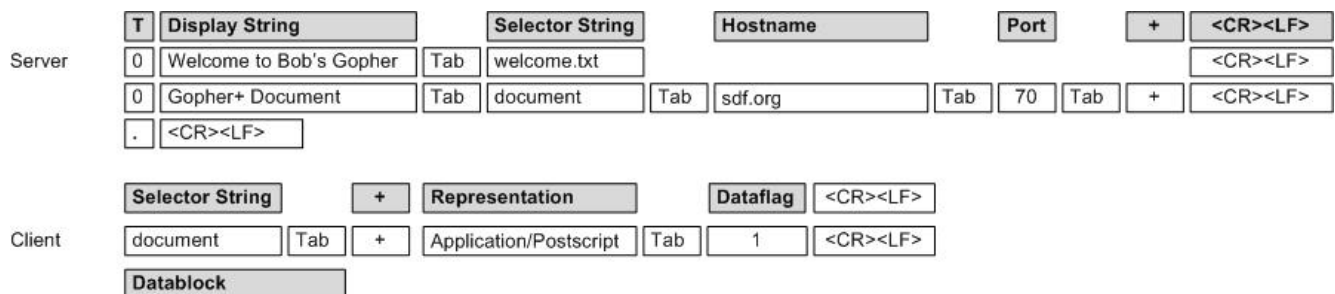


Figure 12 Gopher+ Protocol

Attributes could be access by an information request, “!”, described as, “think of ‘!’ as an upside-down ‘i’ for ‘information’” (Anklesaria, Lindner, McCahill, Torrey, Johnson, & Alberti, Gopher+, Upward Compatible Enhancements to the Internet Gopher protocol, 1993). If an item was Gopher+ capable, sending an ‘!’ in the format

selector string!**<CRLF>**

yielded all of the Gopher+ information available for the item pointed to by the selector string. An example negotiation is illustrated in Figure 13.

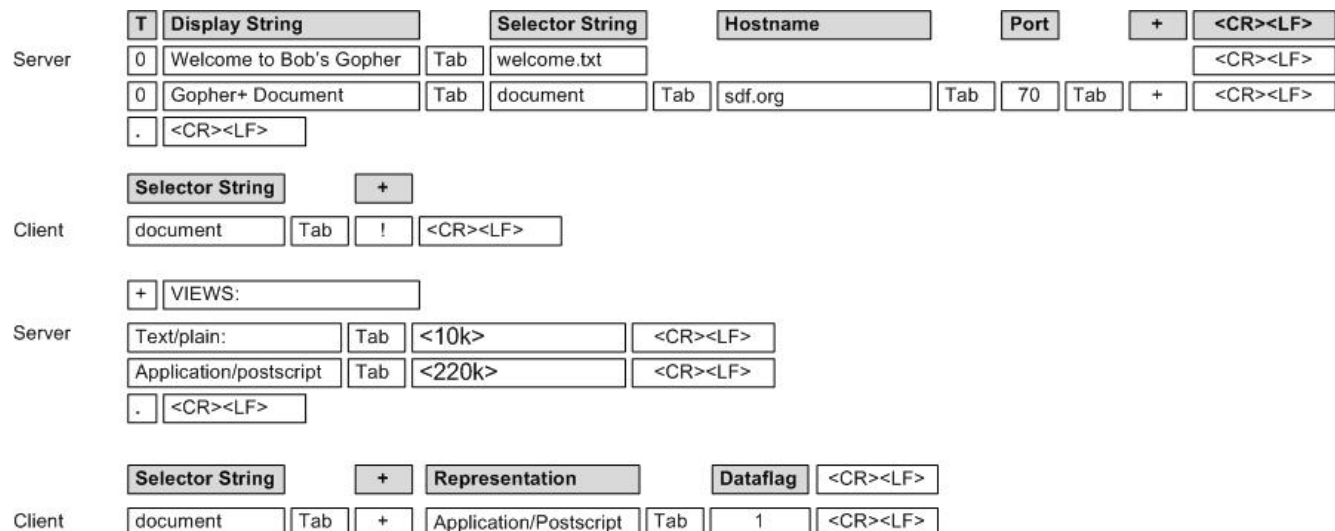


Figure 13 Gopher+ negotiation for Postscript view of a document

In the transaction in Figure 13, the server informs the client that it has a Gopher+ document by including the + at the end of the second line. The client responds with the selector string and a request for “!nformation”. The server responds MIME descriptions of both available views of this document and the size of the items. Any number of views could be available, including foreign-language translations of documents described as “Text/plain De_DE: <15k>” for a German language example.

Attributes were extensible, and initially included

+INFO: A selector string pointing to a Gopher+ document
 +ADMIN: Contact information about the document owner
 +VIEWS: Available alternative views of a document
 +ABSTRACT: Brief multi-line summary of document contents

Interactive Query Items

As with “!”, query items are indicated with a “?”, selection of which results in a form being returned (Figure 14). The “Choose:” query will result in a form with two radio buttons.

Server: T Display String Selector String Hostname Port + <CR><LF>
 0 Welcome to Bob's Gopher Tab welcome.txt <CR><LF>
 0 Gopher+ ASK form Tab questionnaire Tab sdf.org Tab 70 Tab ? <CR><LF>
 . <CR><LF>

Client: Selector String +
 questionnaire Tab ? <CR><LF>

Server: + ASK:
 Note: Slap or tickle? <CR><LF>
 Choose: Which would you like? Tab Slap! Tab Tickle. <CR><LF>
 . <CR><LF>

Figure 14 Gopher+ ASK form

ASK forms required a corresponding back-end script to handle the responses. For example, here are the ASK block and back-end script of a primitive and wildly insecure Mail form:

Mail.ask	Mail.pl
Note: This is the Gopher e-mailing service. Note: Please enter a valid internet e-mail address. Ask: E-mail address: Note: Enter Message AskL:	<pre>#!/usr/local/bin/perl \$Email = <>; chop(\$Name); \$Lines = <>; while(<>) { push(@foo, \$_); } print "Email address is \$Email\r\n"; print "Password is \$Password\r\n\r\n"; print "Message is:\r\n\r\n"; print @foo;</pre>

Available queries include:

ASK	One line text response
ASKP	One line response, characters obscured with bullets (for passwords)
ASKL	Multi-line response
SELECT	Check boxes for none-some-or-all types of responses.
CHOOSE	Radio buttons, only one can be selected
CHOOSEF	Presents a list of files in the local directory to select.

The result of all these efforts was significant: a protocol designed for document retrieval now had the capability of creating and sending documents. As with so much about the early Internet and the dawn of the computing age, getting something to work at all was difficult enough that security was rarely a consideration (Anklesaria, Lindner, McCahill, Torrey, Johnson, & Alberti, Gopher+, Upward Compatible Enhancements to the Internet Gopher protocol, 1993).

Gopher was originally designed as an essentially anonymous document retrieval protocol to facilitate easy access to information rather than limited access. Various kinds of restrictive mechanisms have been implemented at the server end (for example, access restriction by source IP address); however if you have sensitive information, we emphasize that putting it under a Gopher's nose is not a good idea.

At the remove of twenty years the mechanisms in Gopher+ appear woefully insecure – however at the time it was nothing short of astonishing that they worked at all.

But work they did: Gopher+ was considerably more useful than Gopher. Here is one such, a Gopher+ based Usenet query tool, designed to help locate specific information in the vast daily Usenet database (Figure 15) accessed with Matjaž Mešnjak's Windows Gopher+ client available from <http://sites.google.com/site/matjaz85/gopherclient>

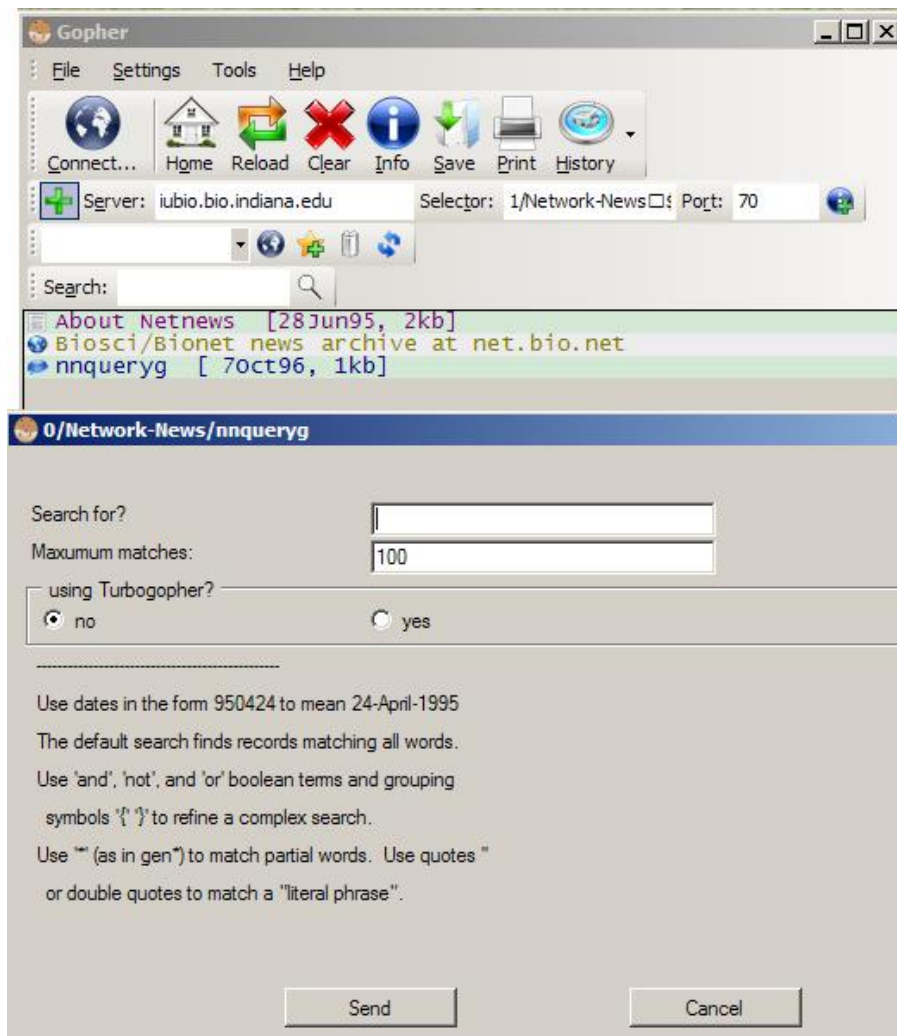


Figure 15 Gopher+ ASK example

Cultural Challenges

Cultures and cultural change had an enormous impact on the fate of Internet Gopher and its participants. We have already seen how the University of Minnesota's culture of conservatism and bureaucracy led to a weak institutional commitment to the groundbreaking protocol that bore the name of its mascot. This bureaucracy and weak commitment has never changed (see Post Script).

And we have touched briefly upon the Internet culture of the early 1990's, when ".edu" domains dominated, and ".com" domains were considered unusual and somewhat crass. These challenges came together during 1993 and 1994 to help doom Gopher.

Civil Service

The University of Minnesota, like many other public institutions, has always offered lower pay for the same positions than in the private sector (Glassdoor, 2010). The response to that complaint, at least during my tenure, was that civil service employees had greater job security than their private sector counterparts. Indeed, it seemed to me that the threat of losing one's job was the primary bogeyman used to keep University employees from leaving for the private sector or asking for raises.

This strongly affected the ambitions of my colleagues, and their attitudes towards “taking Gopher private” (Romenesko, 1996):

The Gopher developers didn't make any money other than their salaries for coming up with what was then the world's hottest Internet application. They discussed leaving the university and starting up a software outfit on their own, but the talks never went anywhere.

"I think it was a little too early and the market wasn't there yet, but I might be wrong," says Torrey

Daniel Torrey does have a point that “the market wasn’t there yet,” for several reasons, not the least of which being the Internet culture at the time that eschewed commercial exploitation of the Internet (more on that in the next section). But the real impediment to spinning off Gopher privately was the fear-based civil-service culture that discouraged taking career risks.

And for many of the Gopher team, academia was their intended career: indeed, Anklesaria remains at the University to this day (Anklesaria F. , 2011), while McCahill remained at the U of M for many years before relocating to Duke University in 2007 (Yen, Mark McCahill, Thanks!, 2007) only a month before Shih-Pau Yen retired from the University (Yen, I'm Finished, 2007).

While there is nothing wrong with a career in academia, this commitment by Gopher’s primary authors to academic careers limited the possibilities for Gopher’s growth. And the U of M’s failure to recognize and support the Gopher team was a profound hindrance to its development. Gopher’s support base was limited to being an unfunded side-project for half a dozen University civil servants.

I was rather an outsider to this culture, because not only had I worked in the private sector, I had actually been an entrepreneur. My first company, Gāmbit MultiSystems, created and franchised the world’s first commercial online interactive role-playing game (Alberti, 2010), the ancestor “World of

Warcraft” and the rest of the multi-billion dollar online game industry. So I was possibly the most frustrated of the Gopher authors, because I knew the potential that Gopher represented.

But while they may have dreamt of riches, the Gopher programmers were not receptive to the idea of taking the development effort private. When I presented the idea of privatization to department head Shih-Pau Yen his only response was a look of stunned incredulity, as if the idea of leaving the University, for any reason, were the craziest notion he had ever heard. This is not really a surprising response from someone committed to a career in academia, but as with the response from the CWIS committee, I was naïve enough to not understand the magnitude of the kind of career change I was suggesting for Yen.

NSFnet's Sacrificial Lamb

While Gopher was nurtured in the thin, stony soil of the University of Minnesota's bureaucracy, it also grew in a particular time and place in the history of the Internet. As mentioned, the Internet was at that time funded by the National Science Foundation, and was supported by the connection fees of each institution. The National Science Foundation acceptable use policies forbade the use of the Internet for profit. As McCahill observed, “We were catching the tail end of... ‘It's just for research; don't be doing commercial stuff here.’” (McCahill, 2001).

This is important for a number of reasons relating to Gopher's eventual demise. First, the research culture of the NSFnet encouraged programmers at other institutions to support the development of Gopher (as well as other protocols). The developers of the WAIS search engine, the Archie FTP archive, the Veronica Gopher archive and other Internet tools all worked collaboratively to build and improve the overall Internet toolkit, including Gopher. Uncounted hours either personal time or time funded at other public institutions went into this mutual development.

These efforts were made willingly because the individuals saw the Internet as a publicly-funded research project. Nobody believed anyone could be taking advantage, particularly financial advantage, of the unpaid work of others.

And because of the stingy support offered Gopher by its parent University, Gopher development was dependent upon this support, beginning with its initial outsourcing under prohibition by the CWIS committee. While the Gopher Team wrote all our own code, we received bug reports from the community, discussed feature ideas and worked to integrate with standards and much more.

Communication was over e-mail and Usenet first in the alt.gopher newsgroup and later on comp.infosystems.gopher.

The Gopher team had never, after all, set out to be Internet mavens. We had written the application as a prototype campus service, and were sadly unschooled in the ways of the Internet. In fact, Gopher initially ran on port 150, a value picked completely out of the air. Only when contacted by IANA – the Internet Assigned Numbers Authority - did the Gopher team learn of the formal method of applying for Internet port assignments, and Gopher’s port was switched to 70 (McCahill, 2001). This is why many of the oldest references to Gopher refer to port 150.

Without community feedback and support the Gopher protocol, clients, servers, and related applications such as gateways and applications would not have been produced as well or as quickly as they were. But the downside to that involvement was that the wider Gopher community felt a strong sense of ownership in the development process. Discussing the demise of Gopher in 2009, one anonymous commenter demonstrated that sense of ownership years later (Mantra, 2009):

To a community that... had contributed bug reports, patches, enthusiasm and good will... had been treated as a peer of the Internet like every other type of organization... the Gopher+ license was deeply insulting and violating. I and most everyone I knew... dropped all work we had been doing or had planned for the Gopher platform.

And this touches upon the conventional wisdom regarding the reason for the demise of Gopher: the decision by Shih-Pau Yen and the University of Minnesota to charge a license fee for commercial users of Gopher software. Indeed, Tim Berners-Lee has repeated this story as recently as 2008, “If we had put a price on [the Web] like the University of Minnesota had done with Gopher, then it would not have expanded into what it is now.” (Waters, 2008).

Certainly the sense of betrayal exhibited by Manta fifteen years later suggests that the licensing issue was extremely controversial, and doubtless it was a factor in Gopher’s demise. To the extent that the licensing issue harmed Gopher, I believe it was due to the investment in Gopher by the wider development community colliding with the changing reality of the end of the NSFnet Internet model. This was a clash of cultures, and the first agency that attempted to bridge the cultural gap, from the

strictly non-profit NSFnet to the inevitable for-profit world was going to pay a high social price for breaking that taboo. As this first agency, Gopher was the “sacrificial lamb” to cultural change.

Admittedly I base this assessment on my own experience. My wife and I are both eldest children, and as a result we served as “icebreakers” for our younger siblings. When, already engaged, we decided to cohabitate before marriage we were met with harsh condemnation particularly from her side of the family. However, when thereafter one of her siblings announced the intention of moving in with a partner before marriage, the news elicited no such condemnation: the world had changed, and having voiced their objections her parents resigned themselves to the new paradigm.

This, I believe, was the same phenomenon that Gopher experienced, exacerbated by its dependence upon the Internet community for development. Already harboring proprietary feelings for Gopher, the development community felt betrayed by the suggestion that the University of Minnesota might profit from “their” software, developed under the strictly nonprofit auspices of the NSFnet.

However, I disagree with the conventional wisdom that the licensing issue was the cause, or even a major cause, of Gopher’s demise. While I agree with Cal Lee that Gopher lost critical “mindshare” over the licensing issue (Lee, 1999), I don’t believe that the licensing controversy was the major factor in Gopher’s demise.

The Overused Phrase “The Perfect Storm” Applies

One reason that I disagree with the conventional wisdom regarding Gopher’s demise is that I don’t think most people realize that Gopher continued to grow and thrive for more than a year following Yen’s announcement of licensing requirements. Plans for licensing were first announced on April 12th at GopherCon ’93 (Riddle, 1993):

SOFTWARE LICENSING: The much-awaited confrontation between Shih-Pau Yen of UMinn and the Gopher community at large was heated, as expected... there were the familiar complaints that the UMinn code wasn't written entirely by UMinn... Mr. Yen seemed determined to proceed with licensing fees of some kind, and it was unclear to me how much he would modify his proposal in the light of the comments at GopherCon.

Despite these objections, and the hurt feelings that persisted for fifteen years in the case of Manta, Gopher continued to grow, as a percentage of Internet packet traffic for more than a year

(Figure 16), and GopherCon '94 was as well-attended as its predecessor. No, other factors came together to result in Gopher's demise in a "perfect storm" of events.

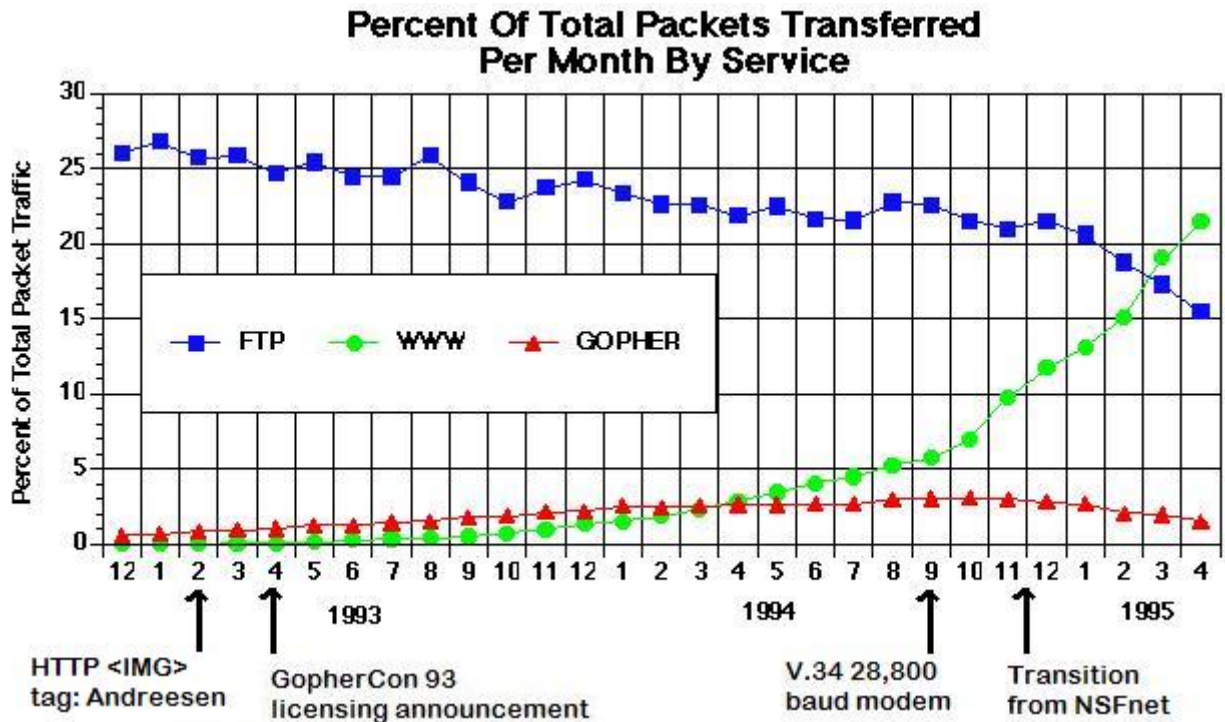


Figure 16 Internet Traffic 1993-1995

Picture This

The first factor that drove the demise of Gopher was the introduction, in 1993, of the IMG image tag into HTML by Marc Andreessen (Andreessen, 1993). Prior to this date, HTML was a text-only system just like Gopher. But why, you may ask, was the IMG tag not introduced until two years after Gopher was created, and six years after the origins of HTML itself? Because neither communication bandwidth nor platform graphic support for images were ready for graphic-laden protocols, whether HTML or Gopher.

Picture Windows

One reason was that the consumer market for graphic computers, and the computers themselves, were still addressing quality and compatibility issues with personal computers,

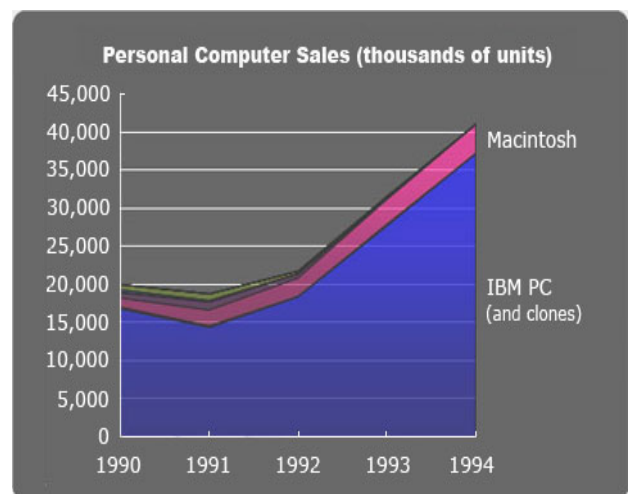


Figure 17 Gopher-era PC Market Growth

particularly the less-expensive IBM PC and Windows platform. While NeXT and Macintosh had provided acceptable graphics quality for several years, their combined market share was merely a fraction of the Windows market (Reimer, 2005).

The total numbers of all type of PC owners boomed during the Gopher era (Figure 17) driving the demand for, and ability to provide, graphic interfaces. By 1993 when Andreessen introduced the IMG tag to the Web, there were an immense new number of computers capable of displaying images.

A Picture is Worth about 10,000 Words

Finally, and most important in my estimation as to why the popularity of Internet Gopher declined, was the introduction in late 1994 of the V.34 28.8K baud modem (Figure 16). This was double the speed of V.33 14.4K baud introduced in 1991 (International Telecommunications Union, 2009). And as the V.34 modems were bundled with booming PC sales, their adoption was rapid.

The doubled modem speed helped push the Web past a critical performance juncture by halving the time that a web page and its accompanying images took to download. The pejorative term “World Wide Wait” was coined by folks watching images fill in slowly on a web page, and early web browsers created an option to disable image loading in order to speed up the user experience. Even with this increase in speed, extensive efforts continued in for years order to improve the Web user experience (Frystyk Nielsen, Gettys, Baird-Smith, Prud'hommeaux, Lie, & Lilley, 1997). But in 1994, V.34 represented about the lowest transmission speed which a patient Web user could endure.

By contrast, Gopher had been written to be extremely speedy: its text-only displays required only a fraction of the bandwidth that a Web page required. Unfortunately what this meant for Gopher was that the improvement in performance between V.33 and V.34 was indiscernibly small. While the Web was achieving an exciting new degree of acceptability, Gopher remained, from the user's point of view, unchanged.

Conclusion

To be sure, other factors contributed to Gopher's demise as well. The NSFnet began its transition in October of 1994 and was retired by April 30, 1995 (NSFnet, 2000). Not coincidentally, Marc Andreessen and James Clark formed Mosaic Communications in December of 1994, with little criticism of this for-profit enterprise by the Internet community. Like my wife's parents, the Internet had become resigned to what was once unacceptable.

In order of significance the factors leading to Gopher's demise were: modems capable of transmitting images at an acceptable speed; the growth in availability of graphically capable personal computers; the advent of post-NSFnet for-profit computing driving the commercial development of graphical Web browsers; the failure of the University of Minnesota to invest in Gopher; and the Gopher licensing controversy. While conventional wisdom claims the latter as the culprit, clear evidence of Gopher's continued growth following the announcement of licensing suggests this is incorrect.

Other minor contributors include the persistent focus by Gopher's advocates on information organizational methods derived from library sciences, and an ivory-tower attitude of disdain for the more free-form and image-intensive Web. While understandable in an academically focused development environment, this dismissal of what the growing user base was demanding – flashy images and magazine style layouts – in favor of a formalized text-only approach only appreciated by information scientists constituted a misinvestment in time and effort that Gopher could ill afford.

Gopher had already been flagging by this time. In 1994, after learning we were expecting our third child I appealed to Yen for a pay increase and was flatly refused. So I left the University and took a job as a Webmaster for a local engineering firm, doubling my salary and committing base Gopher treason. Within a year Gopher usage had dropped dramatically, and the last GopherCon in 1995 was sparsely attended.

Post Script

It was far from lost on me, as I worked on this paper, that April, 2011 was the 20th Anniversary of Internet Gopher. During March I heard someone from the University administration on Minnesota Public Radio, mentioning that it was the tenth anniversary of some medical device that had been developed at the U of M, and so I thought he might be interested to know that April would be the 20th Anniversary of Internet Gopher, so I sent him an e-mail.

Not unexpectedly, my e-mail garnered no response whatsoever.

Six weeks later an old friend who works in Wilson Library on the University's West Bank phoned me up, and told me he had found Gopher in a history display. "It's in Anderson Hall," he said.

This struck me as odd, since 30 years of irregular school and employment at the U of M had not left me with the impression that Anderson Hall had any kind of display area. Anderson Hall is a

peculiar building, a set of large auditoriums set side by side, each serviced by stairs and access door in between, but with no central hallway running its length. Not an idea place for history displays.

I searched the building, and finally phoned up my friend, who left work to help me.

“Not ANDERSON Hall,” he said, “ANDERSEN Hall, across the plaza.”

Only the U of M would build Andersen Hall directly across from Anderson Hall.

We hurried across the plaza, reaching Andersen Hall just as it was closing. Inside was the ‘Headwaters of History’ display (Figure 18). And there, in a display case, placed (fittingly?) next to the first stomach pump³ was a magazine open to a story about Internet Gopher, and featuring a picture of the team. And there I was 20 years and 50 lbs. younger.

Someone at the U of M remembered.



Figure 18 History Display, Andersen Hall

³ It seems odd that Minnesota would be the birthplace of the stomach pump until you remember we also have lutefisk.

References

- Alberti, R. (2010, November 30). *Scepter Project*. Retrieved May 31, 2011, from Scribd: <http://www.scribd.com/doc/45920922/PIL-3252-Scepter-Project-Draft-Final4>
- Andreesen, M. (1993, Feb 25). *Proposed New Tag: IMG*. Retrieved May 31, 2011, from The World Wide Web History Project: <http://1997.webhistory.org/www.lists/www-talk.1993q1/0182.html>
- Anklesaria, F. (2011, May 18). Internet Gopher co-author. (R. Alberti, Interviewer)
- Anklesaria, F., Lindner, P., McCahill, M., Torrey, D., Johnson, D., & Alberti, R. (1993, Jul 30). *Gopher+, Upward Compatible Enhancements to the Internet Gopher protocol*. Retrieved May 31, 2011, from Jumpjet Gopher Archive: <http://jumpjet.info/Gopher/01/Gopher+.txt>
- Anklesaria, F., Lindner, P., McCahill, M., Torrey, D., Johnson, D., & Alberti, R. (March, 1993). *Internet Gopher Protocol*. Retrieved May 31, 2011, from Internet Engineering Task Force: <http://tools.ietf.org/html/rfc1436>
- Berners-Lee, T., Masinter, L., & McCahill, M. (1994, December). *Uniform Resource Locators (URL)*. Retrieved May 31, 2011, from RFC 1738: <http://www.ietf.org/rfc/rfc1738.txt>
- Dalton, M. (1991). Does Anybody Have a Map? Accessing Information in the Internet's Virtual Library. *Electronic Networking*, 1(1), 31-39.
- Dawson, J. (1991, October 21). 'U' Backs Off Plan to Privatize Computer Services. *Minneapolis Star Tribune*, p. 3B.
- Daxial Communications. (2003, Dec 16). *Modem Timeline and Breakdown*. Retrieved May 31, 2011, from Modem Types and Timeline: <http://www.surfthe.us/reference/modem-timeline.html>
- Edmondson-Yurkanan, C. (2002, Jun 11). *The story of FTP's design: 1969-1972*. Retrieved May 31, 2011, from A Technical History of the ARPANET: http://www.cs.utexas.edu/users/chris/think/ARPANET/FTP/tech_story.shtml#spec
- Flowers, S. (1993, Aug 5). Want it? Well Gopher It. *The Guardian (London)*, p. 19.
- Frana, P. (2004, Jan-Mar). Before the Web There Was Gopher. *IEEE Annals of the History of Computing*, Vol. 26(1), 20-41.
- Freed, N., & Borenstein, N. (1996, November). *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Retrieved May 31, 2011, from Internet Engineering Task Force: <http://tools.ietf.org/rfc/rfc2046>
- Frystyk Nielsen, H., Gettys, J., Baird-Smith, A., Prud'hommeaux, E., Lie, H., & Lilley, C. (1997, Jun 24). *Network Performance Effects of HTTP/1.1, CSS1, and PNG*. Retrieved May 31, 2011, from W3C: <http://www.w3.org/Protocols/HTTP/Performance/Pipeline.html>
- Glassdoor. (2010, Apr 7). *University of Minnesota Employee Review*. Retrieved May 31, 2011, from Glassdoor.com: <http://www.glassdoor.com/Reviews/Employee-Review-University-of-Minnesota-RVW456382.htm>
- Higgins, K. (2008, Jan 15). *Who Invented the Firewall?* Retrieved May 31, 2011, from Dark Reading: <http://www.darkreading.com/security/security-management/208803808/index.html>

- International Telecommunications Union. (2009, May 22). *V.34: A Modem Operating at Data Signalling Rates of Up to 33,600 bit/s*. Retrieved May 31, 2011, from ITU.int: <http://www.itu.int/rec/T-REC-V.34/en>
- Kantor, B., & Lapsley, P. (1986, Feb). *Network News Transfer Protocol*. Retrieved May 31, 2011, from Internet Engineering Task Force: <http://tools.ietf.org/html/rfc977>
- Krol, E. (1992). *The Whole Internet User's Guide & Catalog*. Sebastopol, CA: O'Reilly & Associates, Inc.
- Lee, C. (1999, Apr 23). *Where Have All the Gophers Gone? Why the Web beat Gopher in the Battle for Protocol Mind Share*. Retrieved May 31, 2011, from University of Michigan: <http://www.ils.unc.edu/callee/gopherpaper.htm>
- Leong, L. (1994, Nov 14). A Friendlier Internet. *Business Times (Singapore)*, p. 19.
- Lindner, P. (1992, May 13). *Gopher server... changed to port 70*. Retrieved May 31, 2011, from Google Groups alt.gopher: http://groups.google.com/group/alt.gopher/browse_thread/thread/ad9725e998af62b1/cce2724627019d87
- Lindner, P. (1994, Jan). *Gopher World Tour T-Shirt on MTV*. Retrieved May 31, 2011, from Youtube: http://www.youtube.com/watch?v=dyxIwy1bW_M
- Lottor, M. (1992, January). *RFC1296: Internet Growth (1981-1991)*. Retrieved May 31, 2011, from Internet Engineering Task Force: <http://tools.ietf.org/html/rfc1296>
- Mantra. (2009). *Who Killed Gopher? An Extensible Burder Mystery (Comment)*. Retrieved May 31, 2011, from Reddit.com: http://www.reddit.com/r/programming/comments/6uc82/who_killed_gopher_an_extensible_murder_mystery/
- McCahill, M. (2001, Sep 13). Interview with Mark McCahill. 27. (P. Frana, Interviewer) Minneapolis, MN: Charles Babbage Institute.
- Moore, G. (1965, April 19). Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8), 39-42.
- NSFnet. (1992, June). *The NSFNet Backbone Acceptable Use Policy*. Retrieved May 31, 2011, from The Living Internet: http://www.livinginternet.com/doc/merit.edu/acceptable_use_policy.htm
- NSFnet. (2000, Apr 13). *Merit's History: the NSFNet Backbone Project, 1987-1995*. Retrieved May 31, 2011, from NSFNET Transition: <ftp://ftp.ca.freebsd.org/pub/docs/nsfnet/transition/index.html>
- O'Hanlon, A., & Starr, J. (1994, Aug 25). God Is in The E-Mail. *Washington Post*, p. C7.
- Postel, J. (1980, June). *File Transfer Protocol*. Retrieved May 31, 2011, from Internet Engineering Task Force: <http://tools.ietf.org/html/rfc765>
- Reimer, J. (2005, Dec). *Total Share: 30 Years of Personal Computer Market Share Figures*. Retrieved May 31, 2011, from ArsTechnica: <http://arstechnica.com/old/content/2005/12/total-share.ars/7>
- Riddle, P. (1993, May 11). *Trip Report: 1993 GopherCon*. Retrieved May 31, 2011, from PrentissRiddle.com: <http://prentissriddle.com/trips/gophercon1993.html>

- Romenesko, J. (1996, Mar 4). The Death of Gopher. *St. Paul Pioneer Press*, pp. B1, B3.
- Waters, D. (2008, Apr 30). *Web in infancy, says Berners-Lee*. Retrieved May 31, 2011, from BBC News: Technology: <http://news.bbc.co.uk/2/hi/technology/7371660.stm>
- Watson, R., & Barry, J. (1995, Feb 27). When Words Are the Best Weapon. *Newsweek*, p. 36.
- Williams, M. (1995, Jan 25). Help for Beginners Lost in Cyberspace. *The Age (Melbourne)*, p. 30.
- Yen, S.-P. (2007, May 9). *I'm Finished*. Retrieved May 31, 2011, from In My Opinion: <http://www.tc.umn.edu/~yen/opinions/opdocs/finished.html>
- Yen, S.-P. (2007, Apr 9). *Mark McCahill, Thanks!* Retrieved May 31, 2011, from In My Opinion: <http://www.tc.umn.edu/~yen/opinions/opdocs/markthanks.html>

Appendix – Table of Contents

Internet Gopher	1
Introduction.....	2
Rapid Growth	2
Enter the Gopher	3
Outsourced	4
History	5
Setting	6
Structure of the Gopher protocol.....	9
Stateless.....	10
Links	11
Modes.....	11
Anonymous	11
Abstraction	11
The Heyday of Gopher	12
Crystallizing the Solution	12
Challenges.....	12
Publicity	14
Gopher+	14
Cultural Challenges	18
Civil Service.....	19
NSFnet’s Sacrificial Lamb.....	20
The Overused Phrase “The Perfect Storm” Applies.....	22
Picture This	23
Picture Windows.....	23
A Picture is Worth About 10,000 Words.....	24
Conclusion	24
Post Script	25